



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

# IJRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 4, April 2023

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

Impact Factor: 8.423

 9940 572 462

 6381 907 438

 [ijirset@gmail.com](mailto:ijirset@gmail.com)

 [www.ijirset.com](http://www.ijirset.com)

# Interactive CApp using MERRN Stack

Adithya Urs M R<sup>1</sup>, Udit Pathak<sup>2</sup>, Dhanushkamath S<sup>3</sup>, Neha Mahesha<sup>4</sup>, Sunena Rose M V<sup>5</sup>

Engineering Student, Dept. of CSE-AI, Faculty of Engineering and Technology – Jain (Deemed-to-be-University),  
Bengaluru, Karnataka, India<sup>1,2,3,4</sup>

Assistant Professor, Dept. of CSE-AI, Faculty of Engineering and Technology – Jain (Deemed-to-be-University),  
Bengaluru, Karnataka, India<sup>5</sup>

**ABSTRACT:** Students should be aware of what is happening at their college. Students have two options: either they travel around the college keeping track of activities, papers, tests, organizations, etc. and gathering information about them, or they utilize an app that gives them access to all the information, including news and details about their jobs. Students save time by not having to invest time acquiring the necessary information and may instead get it with just one click to better understand their university. This paper is based on our project, which includes a backend and a frontend. The frontend app will be built and developed using React Native, as it allows developers to create apps for both iOS and Android platforms. The backend will be built using new technologies that are faster, more capable, and more productive than traditional systems. The backend will be built using NodeJS and the Express framework. The database will be in NoSQL format, which can manage data dynamically, and a REST API will allow applications to communicate with the backend. It will be hosted by a cloud service provider, as it is more capable and affordable. With the implementation of Natural Language Processing, the chatbot brings an interactive user experience that helps students in answering their queries.

**KEYWORDS:** React, Application, Node, Express, React Native, NoSQL, Chatbot, NLP.Js, AWS, MongoDB.

## I. INTRODUCTION

We are moving towards a future where everything is digitised. When information can be gathered about everything in seconds, why should it be different for students and their universities? The aim is to make an application that aims to provide all the information, such as technical events, cultural events, workshops, seminars, and information about their subjects, assignments, presentations, that a student will need in his student life. It also provides information about the companies coming to the college and the packages they offer.

## II. RELATED WORK

This paper[1] is based on an Android app that a university uses to store and manage student information. The application will be created in Kotlin, which Google has designated as the preferred language for developers. The Android Room database will house the application data. The `v7recyclerview` module will be used to create the user interface since it offers high usability and functionality. A smart assistant that enables students to ask questions and receive responses via text and voice from the application has been created using the Alan App in Kotlin.

Paper [2,3] both are based on the College website and portal where the app provides the required information a student should know about their Uni or College. The application is built using the latest MERN stack.

Paper[4] is based on the Chatbot and its applications. The chatbot software assists students in learning about the college admissions process and receiving prompt responses wherever there is an online connection. These chatbot systems minimize the workload of the admissions office by giving the information that students or parents require while simultaneously decreasing the workload of the department by responding to every message that students ask for.



### III. METHODOLOGY

Two components make up our project: a user-facing mobile application and an admin panel (students). The admin panel is used to upload, create, or delete any type of data that users would be able to see on the mobile application. A single API server created using Express connects both project components.

#### ADMIN PANEL

The aim of the admin panel, which was created with React and the Express server, is to add data to the database. Only particular users will be granted access to the admin panel, which will be restricted. Role-based authentication has been used to implement this application. So, only certain actions that are permitted for that specific role can be performed by a specific user with that role. Certain people have access to the admin panel to add, alter, or delete students, events, clubs, firm placements, hackathons, etc. Network API calls from the React app to our API server are used to carry out these operations. From that point on, all actions are committed to and communicated with MongoDB.

#### MOBILE APPLICATION

React Native was used to create the mobile application since it makes it easier to create apps that run on both Android and iOS with the same code. The purpose of this mobile application is to provide students with access to a variety of information on their academics, job placements, events, hackathons, etc. The application performs obtain operations from the database by sending network API requests to our API server. The admin panel's API is also utilized by this application.

#### REST API's

Express was used to create our REST API in the Node environment. Both the admin panel and the mobile application use this API server as a channel for communication with the database. As it is so crucial to both applications, it serves as the project's brain. As this needs to be scalable, we had to choose Express Node above other technologies because it's very simple to scale in terms of increasing user numbers and API requests. We only need to add a load balancer to the system to spread a load of requests among the nodes in order to scale our REST API. We can do this by utilizing Kubernetes or any other coordinating platform to generate numerous containers. This would facilitate scaling horizontally up to a significant scale.

#### CHATBOT

The chatbot assists students in answering their queries. With the power of Natural Language Processing, the chatbot analyses the input given phrases and answers them which simulates human conversations. The chatbot is built using the NLP.js package which is a natural language utility built for Node.js. For the UI part, we are using the react-native-gifted-chat package which has built-in chatbot UI components for React Native.

### TECH STACK

FRONT END	React, Redux, React Native, Context API
BACK END	Express.js, Node.js, Mongoose, MongoDB
CLOUD SERVICES	AWS EC2 Instances, AWS S3 Buckets, AWS IAM
CHATBOT	NLP.js

Table 1: Technologies employed in this project

#### 1. React

An open-source JavaScript package called React is used to create front-end UI elements [5]. A group of engineers and Meta engineers maintain it. This technology was utilized to create our admin panel. We chose this technology over the

outdated plain HTML and CSS because react offers dynamic page rendering and using Axios and Redux for state management made it very easy to make network calls.

## 2. React Native

React Native is an open-source JavaScript framework for building apps for mobile devices [6], TVs, the web, and other platforms. React Native was used in the development of our mobile application since it allowed us to use a single code base to support both Android and iOS. In order to efficiently maintain the state and make network calls, we also used ContextAPI.

## 3. Express.Js

RESTful APIs can be created using the Express.js framework in Node.js environments. It is entirely open source and is maintained by an engineering community [7]. The REST API of our application is constructed using Express.js. This builds a straightforward server and uses it to deliver our API.

## 4. Node.js

A runtime server environment is Node.js. A V8 JavaScript engine powers Node.js outside of the browser [8]. Node.js has several LTS versions and is entirely open-source. We choose the Node.js runtime environment because of its ability to scale horizontally and because async and await make it simple to make asynchronous network requests.

## 5. Mongo DB

An open-source NoSQL database with a document-oriented structure is called MongoDB [9]. In this, documents are a collection that MongoDB uses to store data in a json-like manner. While MongoDB Inc. has created a cloud-hosted MongoDB database that is both paid and has a free tier, the database is entirely free to use. As MongoDB adheres to the NOSQL structure and is horizontally scalable for future requirements, it serves as our main database.

## 6. AWS

Amazon Web Services is a platform for cloud computing where it offers its resources and charges users on a usage-based basis each month [10]. AWS offers a wide range of services to its consumers, who come from different industries such as businesses, governments, and start-ups. All our frontend and backend applications are hosted on AWS EC2 instances, and all of the photos uploaded through the admin panel are stored in AWS S3 buckets.

## 7. NLP.Js

An open-source JavaScript library for natural language processing (NLP) is called NLP.js [11]. It is capable of carrying out a variety of NLP tasks, including named entity recognition, sentiment analysis, part-of-speech tagging, tokenization, and part-of-speech tagging. The primary differentiating feature of NLP.js is a better user experience that takes advantage of increasing data privacy & security controls and options while improving response time, adding more language support, and, according to certain benchmarks, improving accuracy.

## **IV. PROJECT ARCHITECTURE**

We have implemented a design that will be scalable both horizontally and vertically. React and React Native are used to build the mobile application and the frontend of the admin panel, respectively, to appeal to both Android and iOS users. We used Express.js in a Node environment for the RESTApi. As we expand the number of containers on which we can deploy the node environment with our RESTApi hosted on it utilizing any orchestrating platform, this approach will become increasingly scalable. We will then utilize a load balancer to distribute the load among the containers. We utilized MongoDB for the database management portion since it provides horizontal scalability and adheres to a NoSQL framework. Our project architecture is shown in broad strokes in Figure 1.



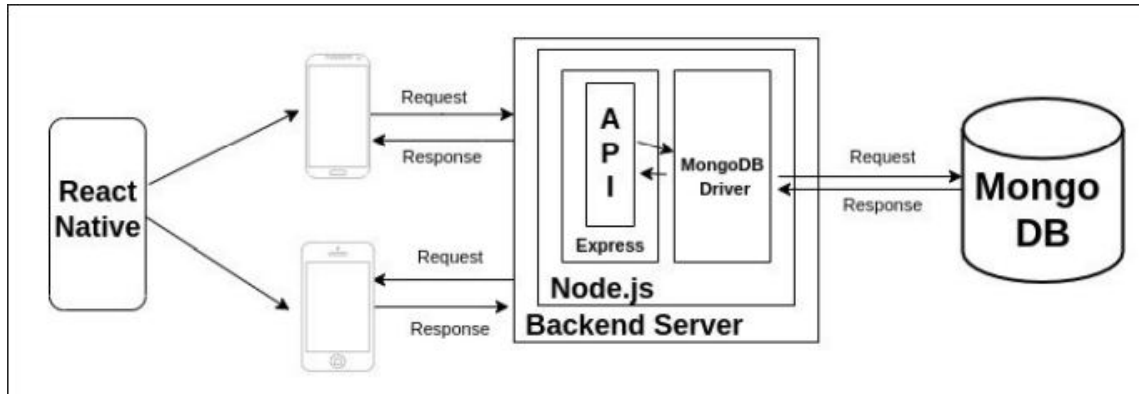


Figure 1: Architecture of this project

The MERRN stack, which stands for Mongo, Express, React, React Native, and Node.js, denotes our entire paper's architecture.

### V. RESULTS

The below figures 2,3, and 4 are the screens for ADMIN Panel, Mobile App and Chatbot

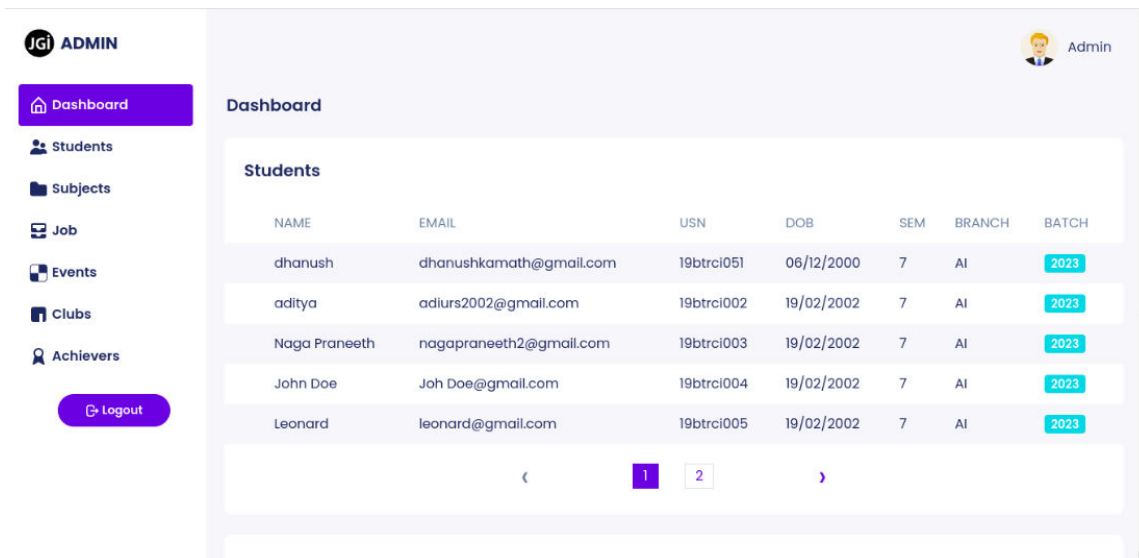


Figure 2: Dashboard of Admin Panel



|| Volume 12, Issue 4, April 2023 ||

| DOI:10.15680/IJIRSET.2023.1204062 |

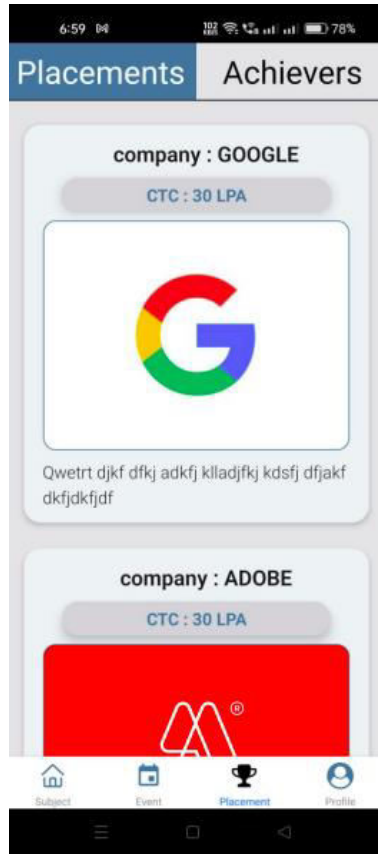


Figure 3: College App – Placement and Achievers Screen

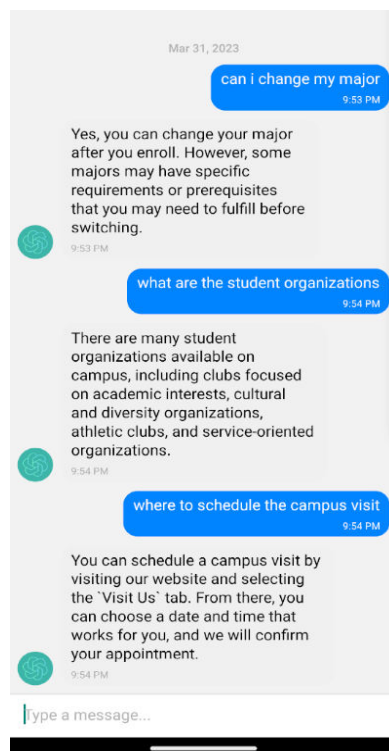


Figure 4: Chatbot screen



## VI. CONCLUSION

The proposed work seeks to use cutting-edge frontend (React Native) and backend (NodeJS, Express, NoSQL) technology as well as a highly scalable architecture (MVC architecture) to create a programme that will simplify the lives of the students by enabling them to learn about all the resources the college offers assisting them in maximizing their time and college experience.

### I. FUTURE WORK

The application we created in this paper allows users to view information whenever they open it. In the future version, push notifications will be enabled, allowing users to be notified whenever new content is posted. Future versions of our application may include features like a schedule of lectures, an application for campus employment, and a help desk that may address simple problems for students.

## REFERENCES

- [1] Ananthi P, Abinaya G, Diwaan Chandar C S, Aarthi K V, & Vasundhara B. (2022, November 28). ANDROID APPLICATION FOR STUDENT MANAGEMENT SYSTEM USING KOTLIN. *International Journal of Engineering Technology and Management Sciences*, 6(6), 88–91. <https://doi.org/10.46647/ijetms.2022.v06i06.013>
- [2] Patil, D. R., Gentyal, V., Mudaliar, V., Kanpurne, G., & Ambi, D. (2022, April 30). College Website Using MERN Stack. *International Journal for Research in Applied Science and Engineering Technology*, 10(4), 1096–1098. <https://doi.org/10.22214/ijraset.2022.41450>
- [3] Saijal Gupta, Gaurav Kumar, Akshaya Renganathan, Pooja and Akshay Jain. Online College Portal using MERN Stack. *International Journal for Modern Trends in Science and Technology* 2021, 7 pp. 338-343. <https://doi.org/10.46501/IJMTST0712066>
- [4] Shingte, Kshitija and Chaudhari, Anuja and Patil, Aditee and Chaudhari, Anushree and Desai, Sharmishta, Chatbot Development for Educational Institute (June 6, 2021). Available at SSRN: <https://ssrn.com/abstract=3861241> or <http://dx.doi.org/10.2139/ssrn.3861241>
- [5] [https://en.wikipedia.org/wiki/React\\_\(JavaScript\\_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))
- [6] [https://en.wikipedia.org/wiki/React\\_Native](https://en.wikipedia.org/wiki/React_Native)
- [7] <https://expressjs.com/en/5x/api.html>
- [8] <https://nodejs.org/en/about/>
- [9] <https://www.mongodb.com/docs/atlas/>
- [10] [https://docs.aws.amazon.com/ec2/?icmpid=docs\\_homepage\\_featuredsvcs](https://docs.aws.amazon.com/ec2/?icmpid=docs_homepage_featuredsvcs)
- [11] [https://docs.aws.amazon.com/ec2/?icmpid=docs\\_homepage\\_featuredsvcs](https://docs.aws.amazon.com/ec2/?icmpid=docs_homepage_featuredsvcs)



**INNO SPACE**  
SJIF Scientific Journal Impact Factor  
**Impact Factor: 8.423**



**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

**9940 572 462** **6381 907 438** **ijirset@gmail.com**



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details